1. Write a C Program to reverse a string without using an additional library functions.

#include <stdio.h>

void revStr(char\* str)

{

int length = 0;

int i;

char temp;

for (length = 0; str[length] != '\0'; length++);

for (i = 0; i < length / 2; i++)

{

temp = str[i];

str[i] = str[length - i - 1];

str[length - i - 1] = temp;

}

}

int main()

{

char str[100];

printf("Enter a string: ");

scanf("%s", str);

revStr(str);

printf("Reversed string: %s\n", str);

return 0;

}

1. Explain the concept of structures in C and write a program to store student information (name, roll number, marks) using a structure.

A structure is defined using the struct keyword followed by the structure name and a block of variables (called members). Each member can be of any data type and is accessed using the dot (.) operator.

#include <stdio.h>

struct Student {

char name[50];

int rollNumber;

float marks;

};

int main() {

struct Student student;

printf("Enter student name: ");

scanf("%s", student.name);

printf("Enter roll number: ");

scanf("%d", &student.rollNumber);

printf("Enter marks: ");

scanf("%f", &student.marks);

printf("\nStudent Information:\n");

printf("Name: %s\n", student.name);

printf("Roll Number: %d\n", student.rollNumber);

printf("Marks: %.2f\n", student.marks);

return 0;

}

1. Explain the concept of pointers in C and write a program to swap the values of two variables using pointers.

Pointers are variables that store the memory address of another variable. They provide a powerful way to directly access and manipulate memory, which is crucial for tasks like dynamic memory allocation, efficient array handling, and implementation of data structures such as linked lists.

#include <stdio.h>

void swap(int \*a, int \*b) {

int temp;

temp = \*a;

\*a = \*b;

\*b = temp;

}

int main() {

int x, y;

printf("Enter value for x: ");

scanf("%d", &x);

printf("Enter value for y: ");

scanf("%d", &y);

printf("Before swapping: x = %d, y = %d\n", x, y);

swap(&x, &y);

printf("After swapping: x = %d, y = %d\n", x, y);return 0;

}

1. **Implement a function in C that takes an integer array and its size as arguments and sorts the elements of the array in ascending order using a selection sort algorithm.**

#include <stdio.h>

void selectionSort(int arr[], int size) {

for (int i = 0; i < size - 1; i++) {

int minIndex = i;

for (int j = i + 1; j < size; j++) {

if (arr[j] < arr[minIndex]) {

minIndex = j;

}

}

if (minIndex != i) {

int temp = arr[minIndex];

arr[minIndex] = arr[i];

arr[i] = temp;

}

}

}

int main() {

int arr[100], size;

printf("Enter the number of elements: ");

scanf("%d", &size);

printf("Enter the elements:\n");

for (int i = 0; i < size; i++) {

scanf("%d", &arr[i]);

}

selectionSort(arr, size);

printf("Sorted array in ascending order:\n");

for (int i = 0; i < size; i++) {

printf("%d ", arr[i]);

}

printf("\n");

return 0;

}